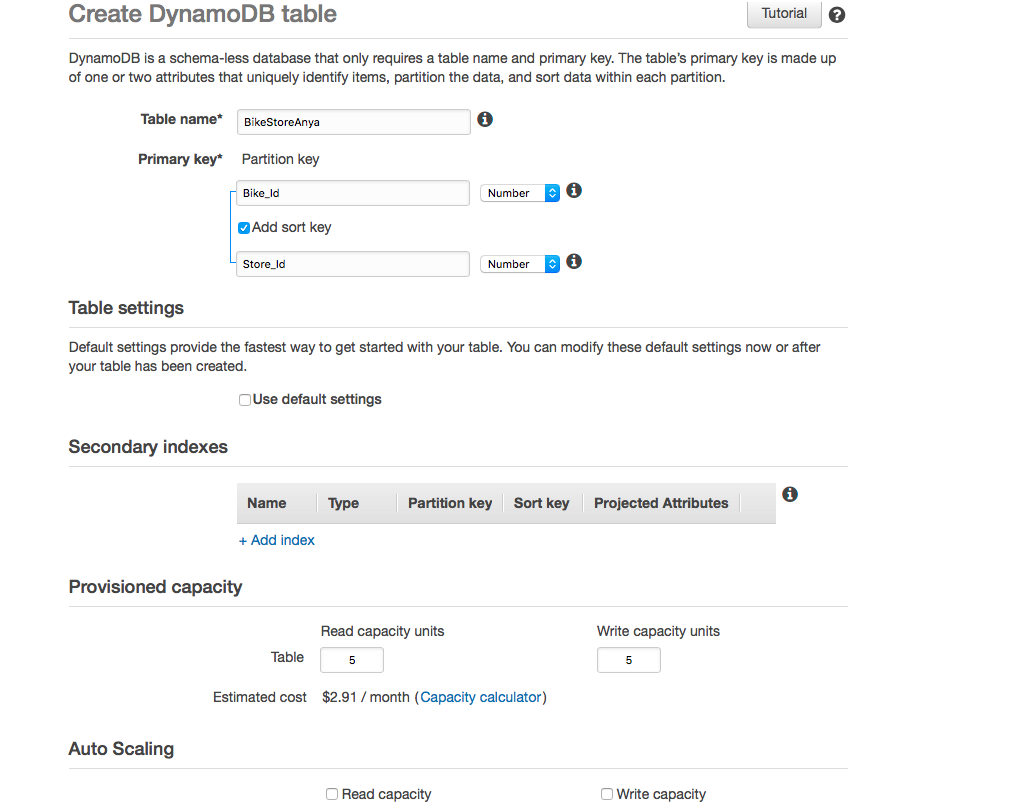
**Lab – DynamoDB Week 2**

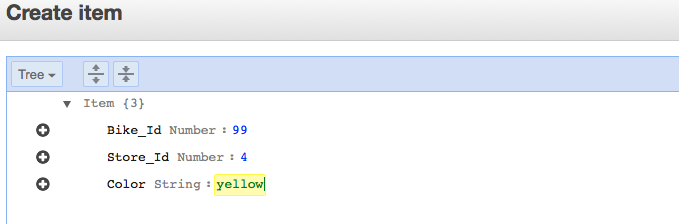
**Exercise 1**

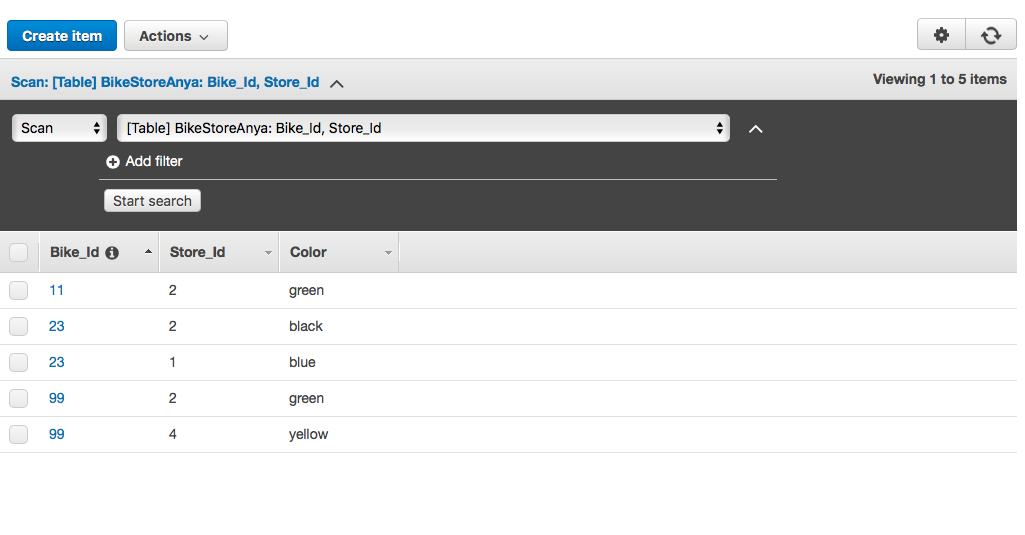
**Learning Objective:**

1. Observe differences between Partition vs Sort keys
2. Observe differences between Query and Scan
3. Create table **BikeStoreYOURNETID**
4. Add Primary Key – Bike\_Id (Number)
5. Add Sort Key – Store\_Id (Number)
6. Uncheck Use Default Settings
7. Uncheck Read Capacity and Write Capacity
8. Click Create

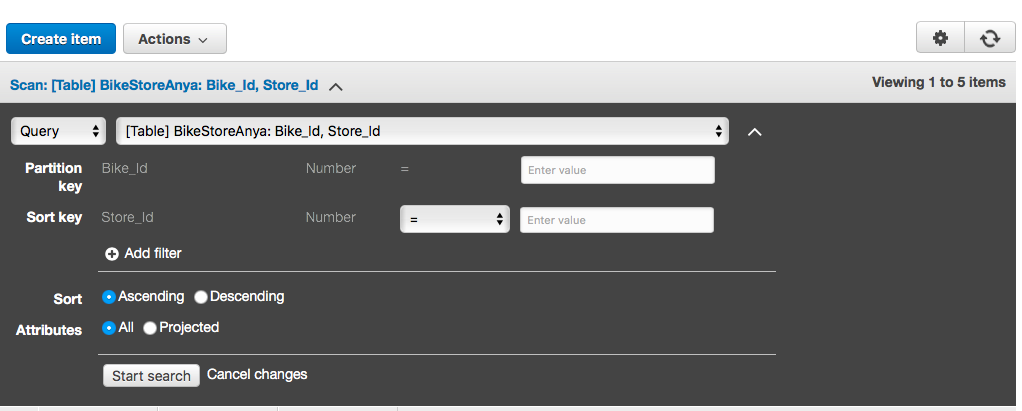


1. Enter 5 different items to **BikeStoreYOURNETID** table as follows:
2. Bike\_id = 99; Store\_Id = 4; color (string) = yellow
3. Bike\_id = 23; Store\_Id = 1; color (string) = blue
4. Bike\_id = 23; Store\_Id = 2; color (string) = black
5. Bike\_id = 99; Store\_id = 2; color (string) = green
6. Bike\_id = 11; Store\_id = 2; color (string) = green

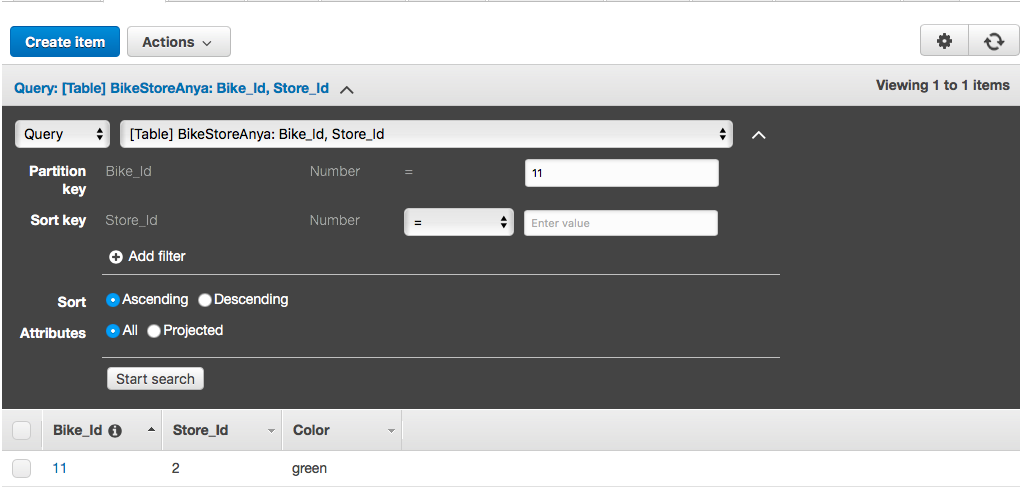




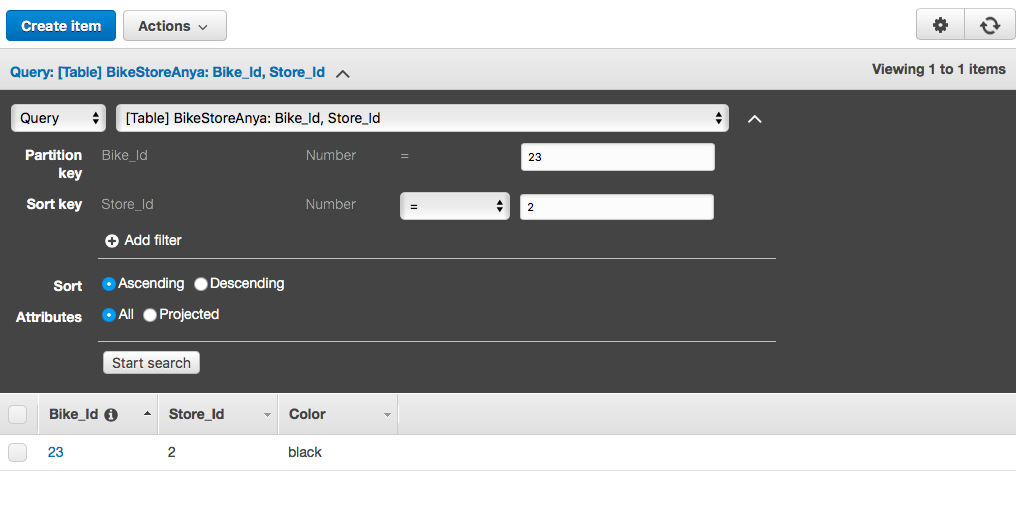
1. Let’s look at Query Option (based on Partition and Sort Key)
2. Choose Query from drop down list



1. Observe that for Partition key can only query for specific value (chose 11 for demonstration)



1. Observe that with Sort Key we can do more advanced lookup. Enter 23 for Partition Key and 2 for Sort Key:

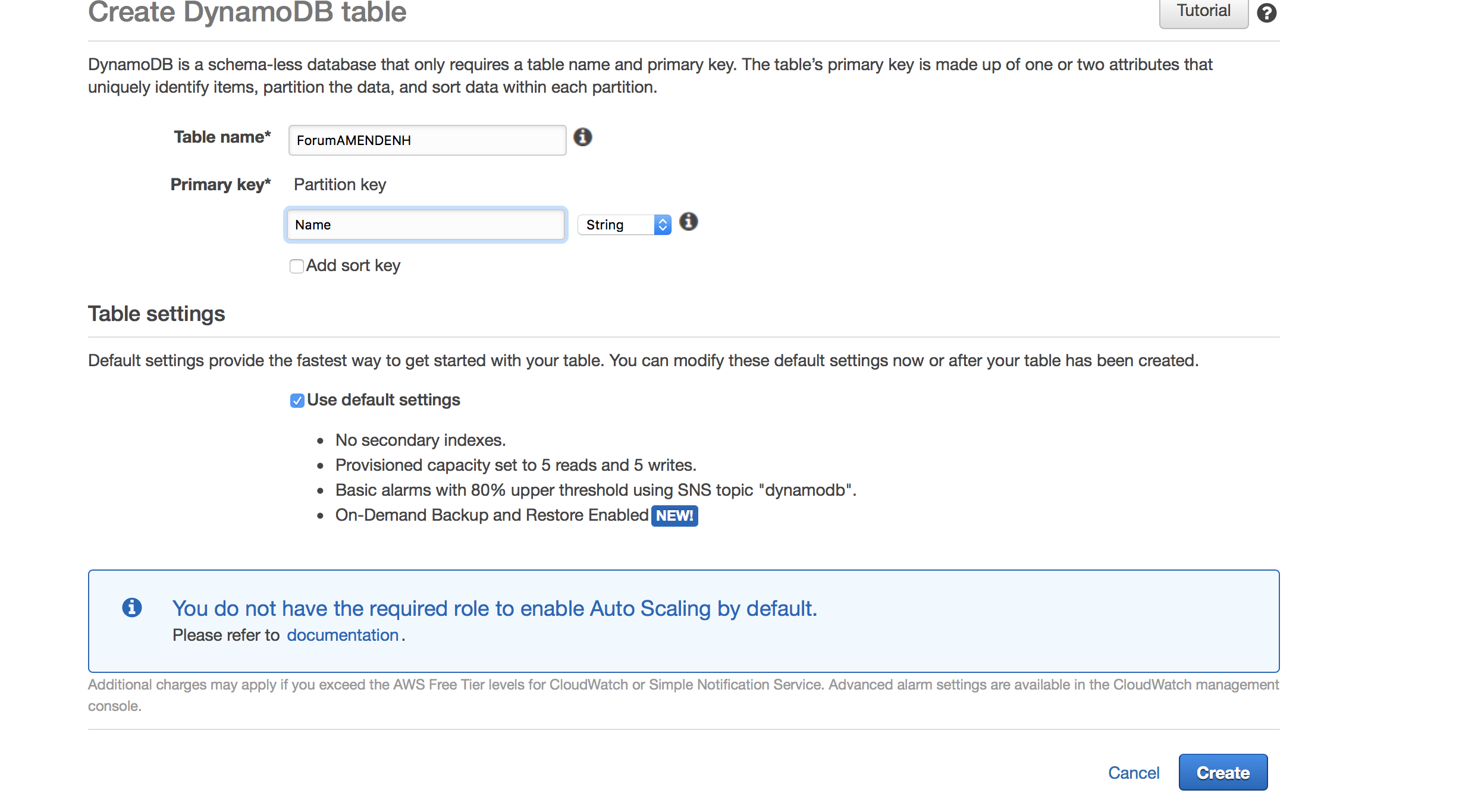


1. Exit out of table view

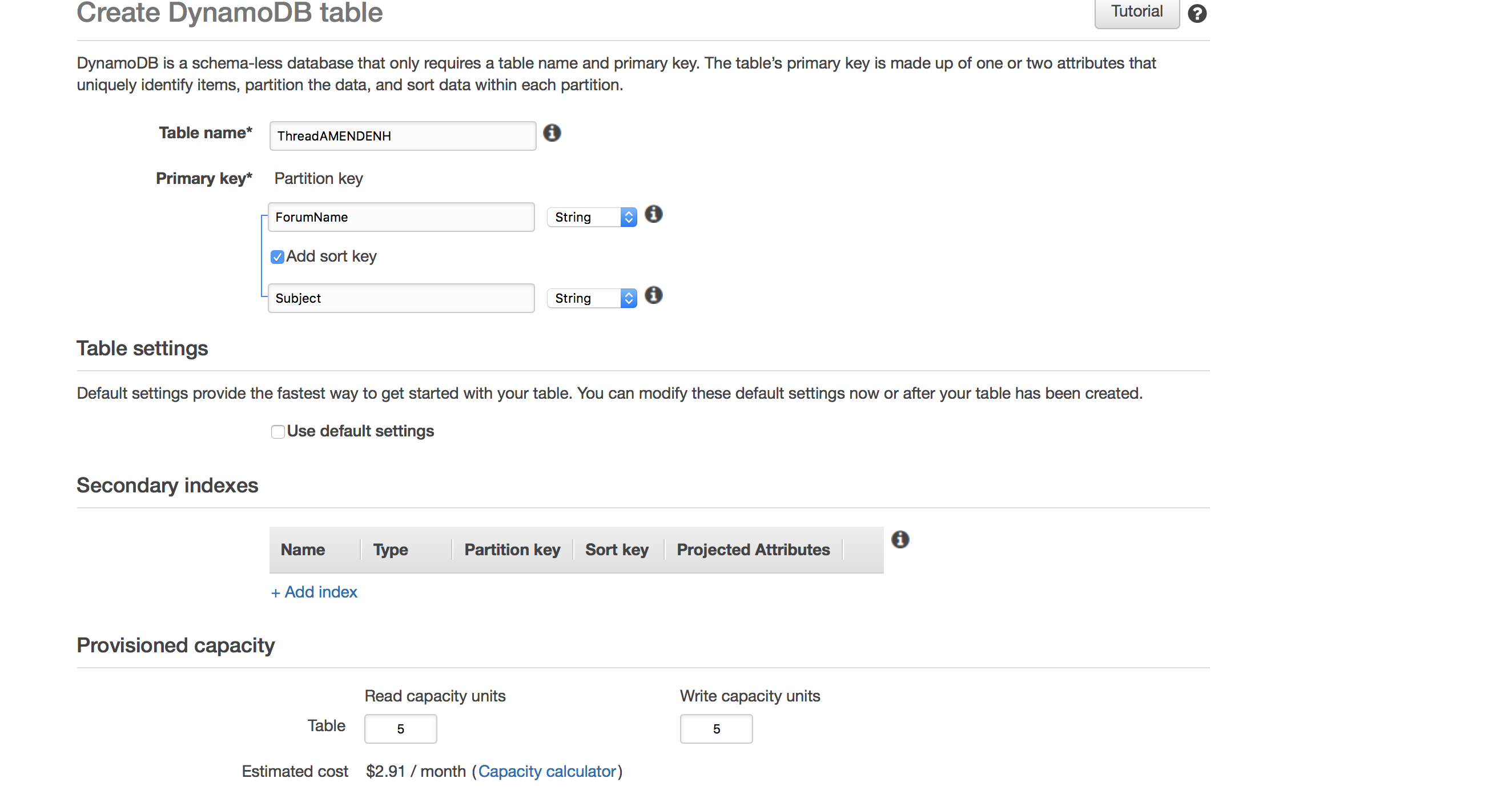
**Exercise 2**

**Learning Objective**

1. Create 2 tables in DynamoDB (Forum and Thread)
2. Create Local Secondary Indexes on Thread table
3. Insert test data in both tables
4. Query and Scan Items
5. Create table **ForumYOURNETID** as follows:
6. Table Name: Enter **ForumYOURNETID**
7. Primary Key: Enter **Name** and set type to **String**
8. Uncheck default settings
9. Uncheck Read and Write Capacity
10. Click **Create** to create this table

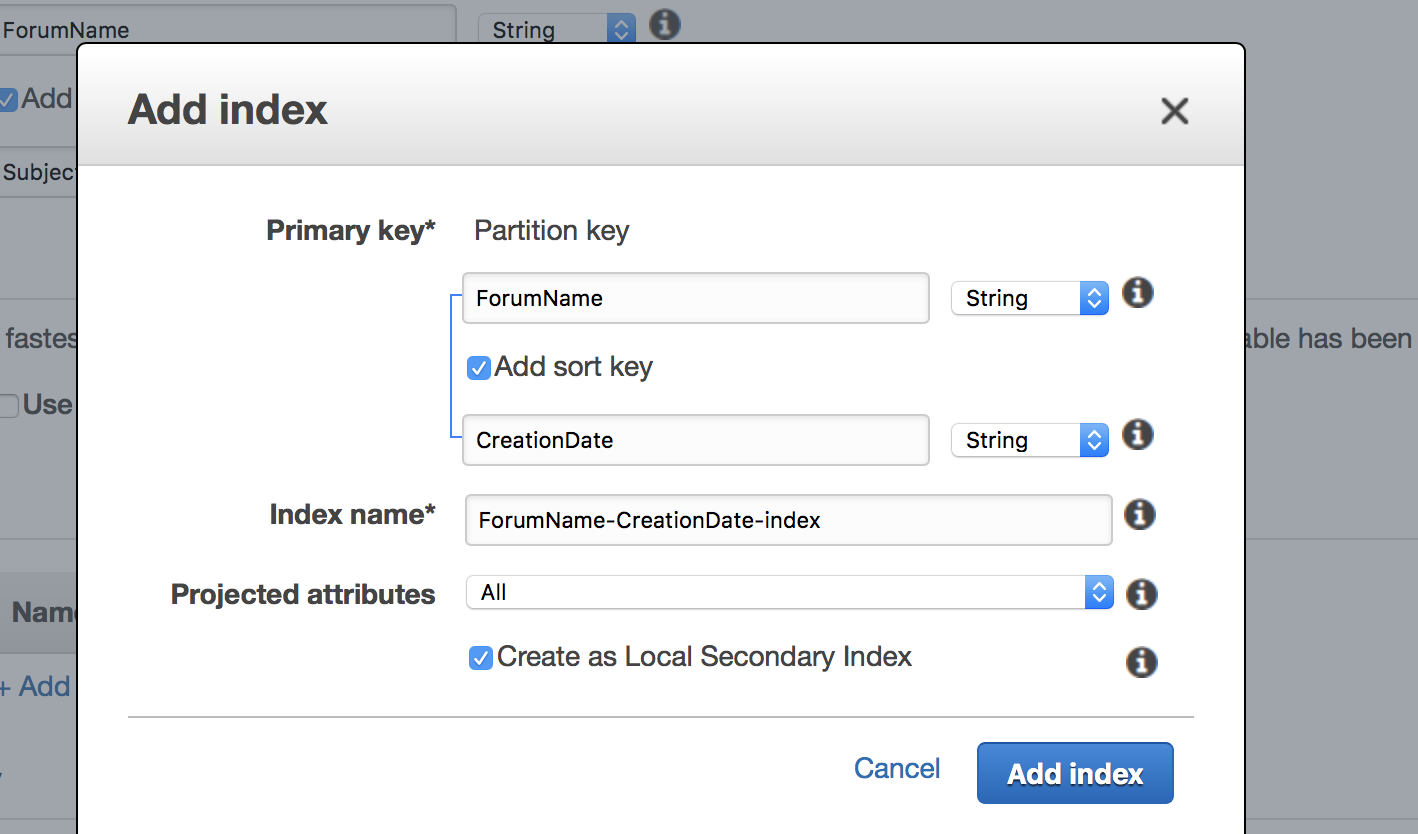


1. Create second table **ThreadYOURNETID** , add Sort Key and Indexes to it:
2. Table Name: ThreadYOURNETID
3. Primary Key: Forum Name of type String
4. Check Add sort key
5. Sort Key: Enter Subject and set the type to String

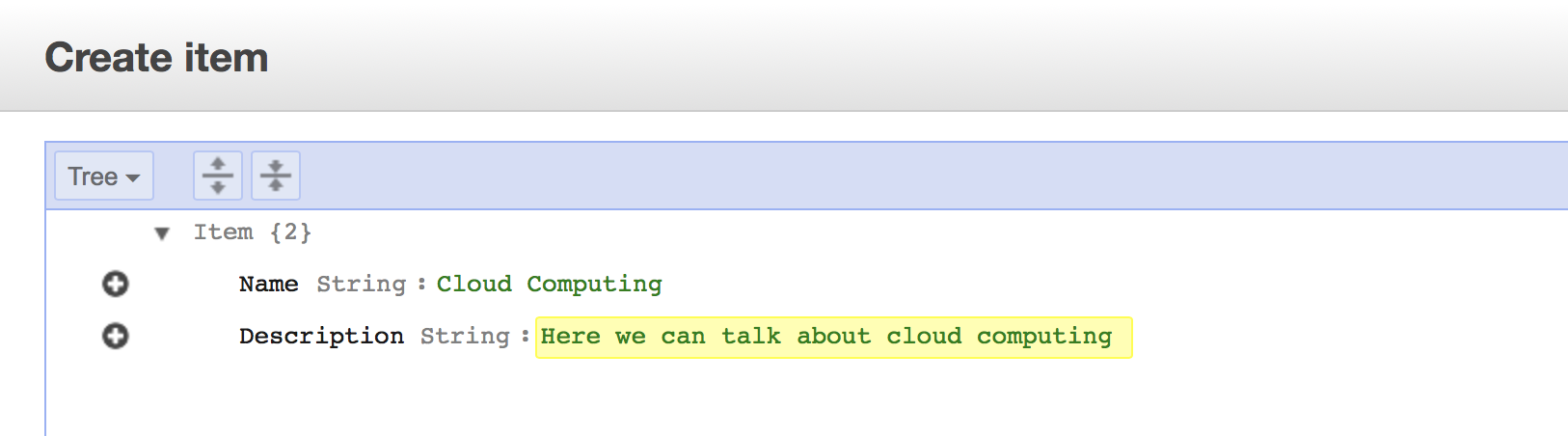


1. Click **+Add Index** in the **Secondary Indexes** section and then create Local Secondary Index for Thread table with following parameters:

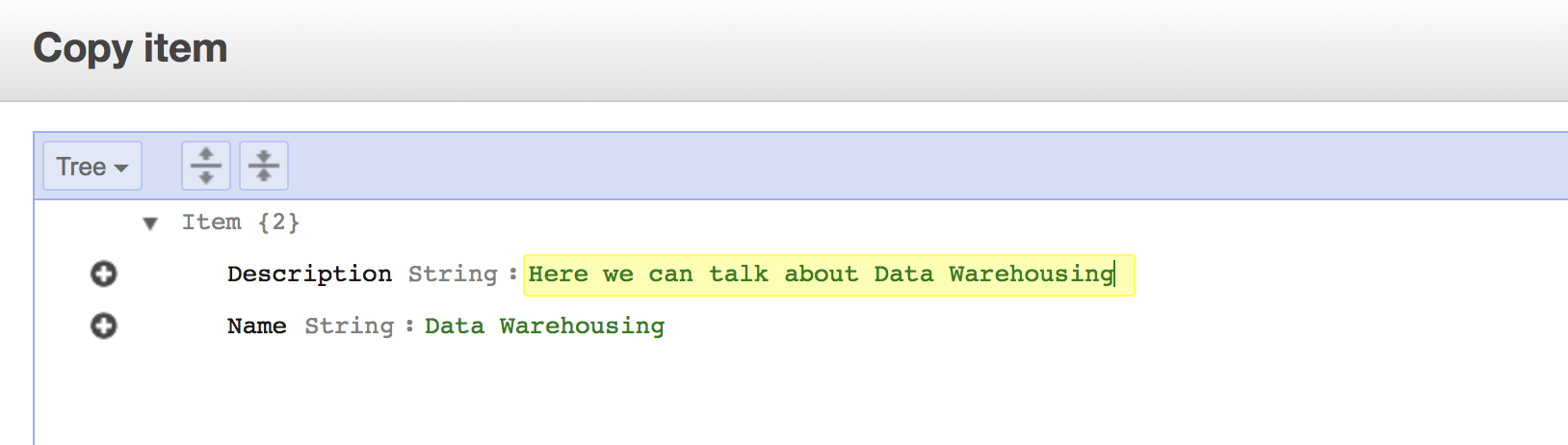
* **Primary Key**: Enter ForumName and set type to String
* **Sort Key**: Enter CreationDate of type String
* **Index Name**: Enter CreationDate-Index
* **Projected Attributes**: All Attributes
* **Index Type**: Make sure to check Local Secondary Index box



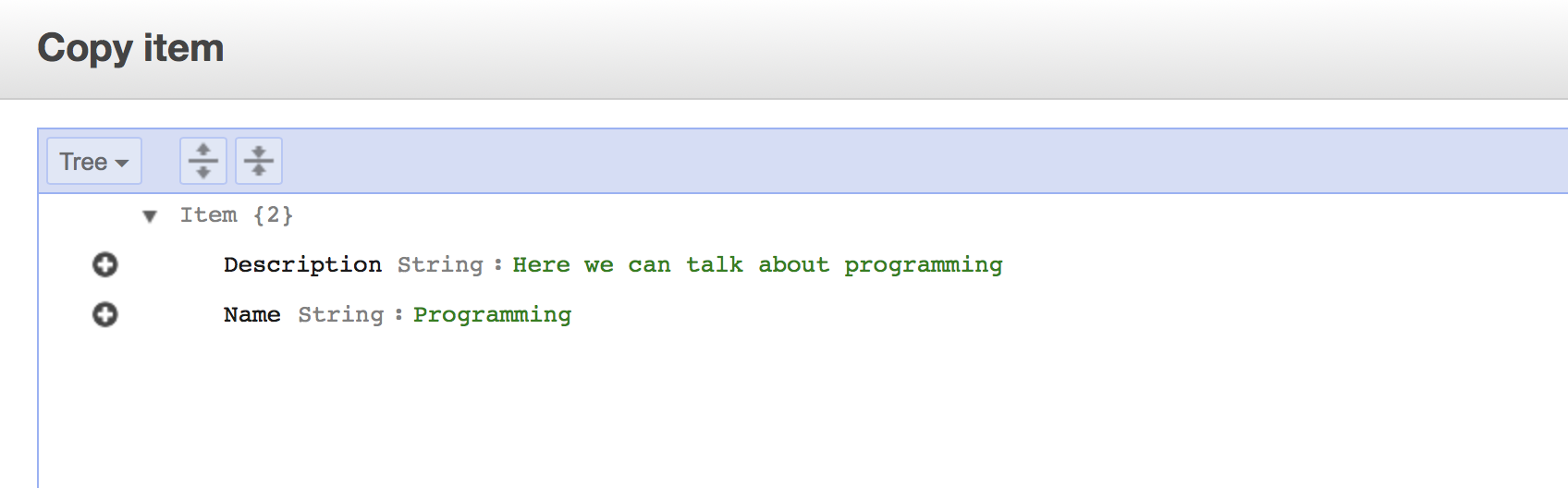
1. Click Create to create this table
2. **Insert test data into ForumYOURNETID table (add 4 items)**
3. Open Forum table and click Create first Item as follows and save:



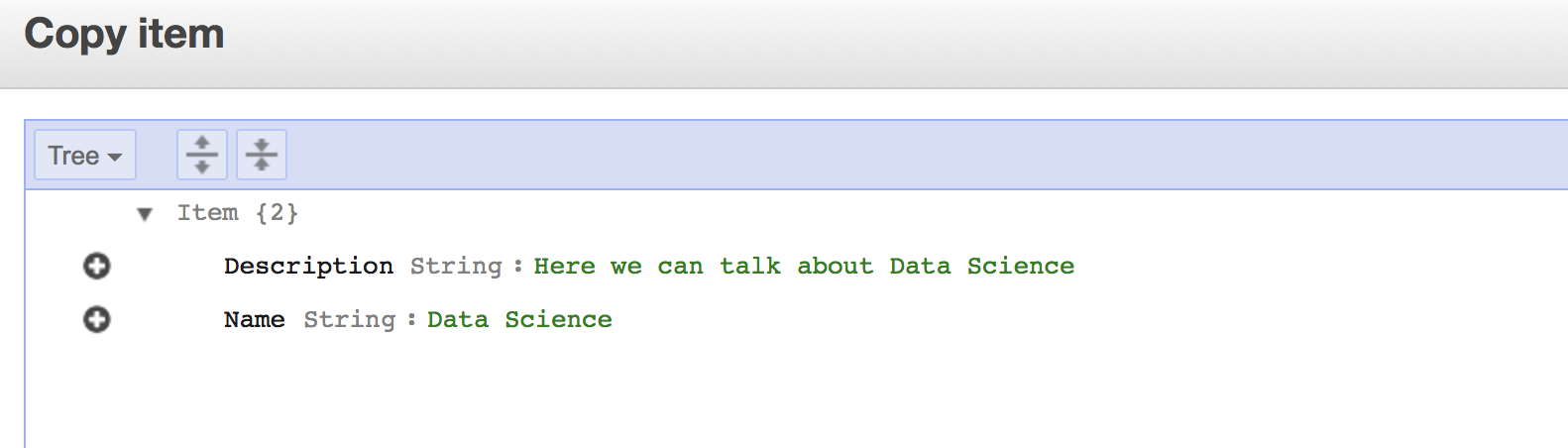
1. Add second item as follows and save (use Duplicate feature for faster creation):



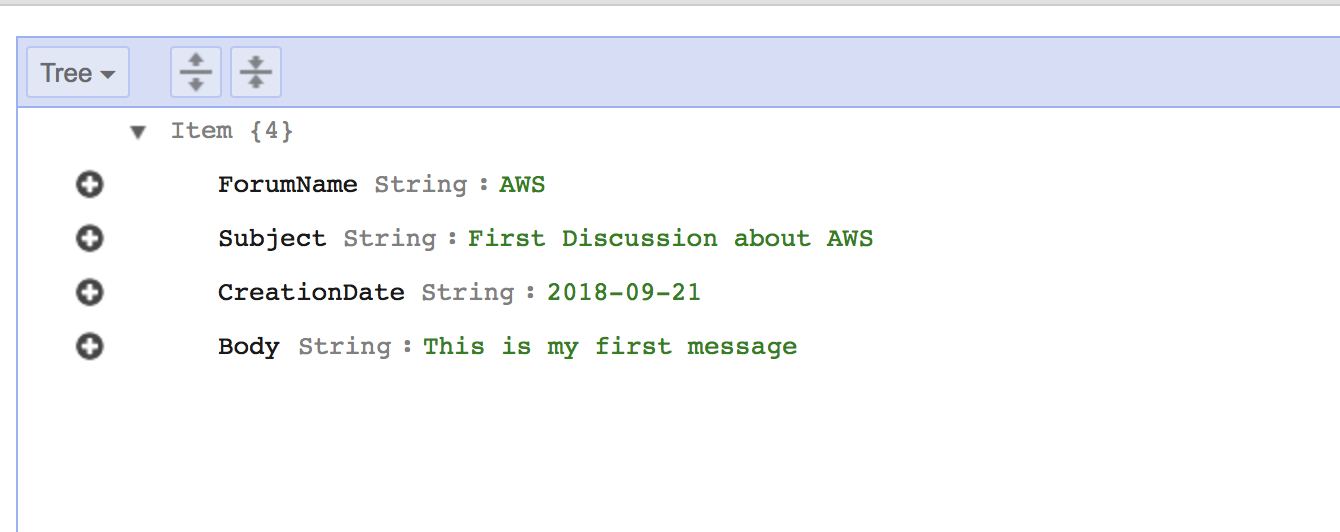
1. Add third item as follows and save:



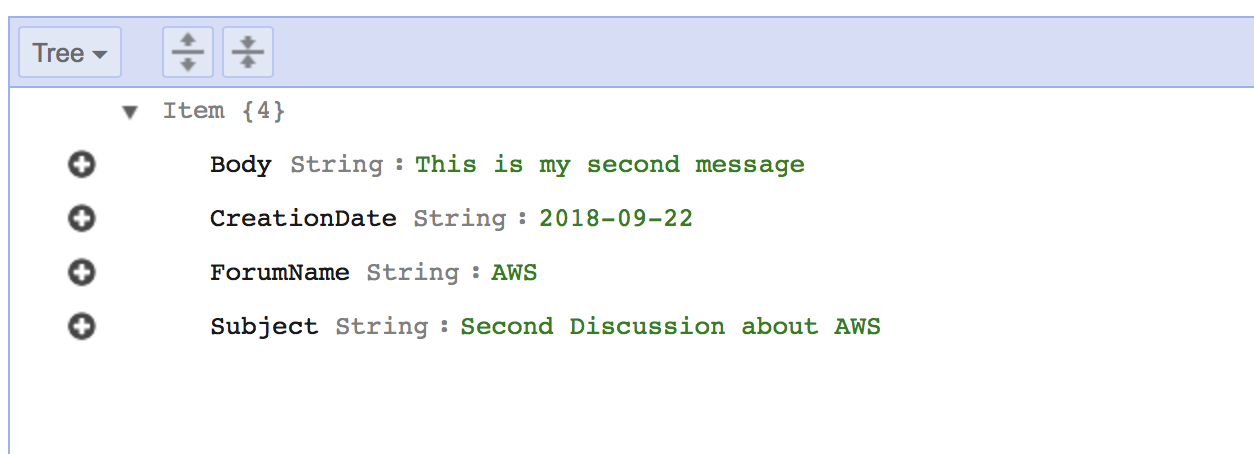
1. Add forth item as follows and save:



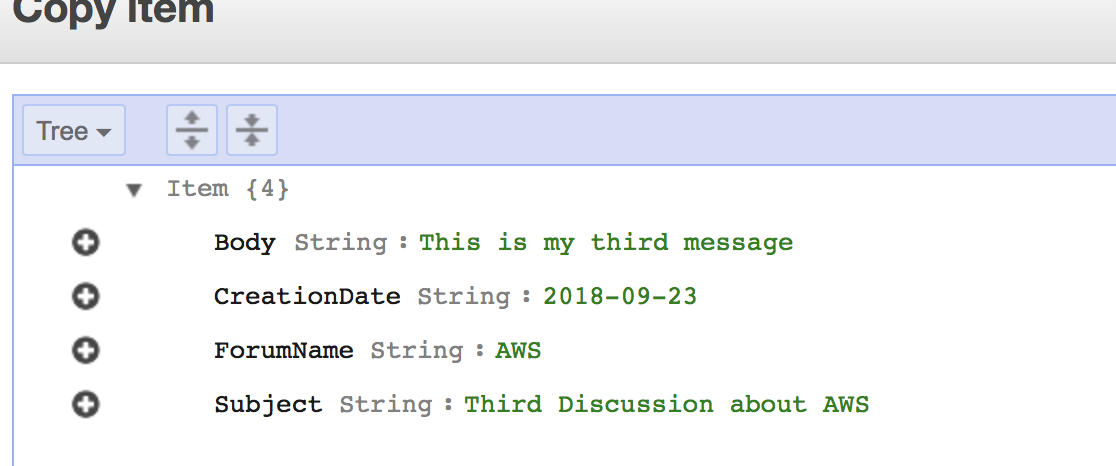
1. Insert data into **ThreadYOURNETID** table (4 items) and save. The Thread table is “partition and sort” with **CreationDate** as Local Secondary Index. To create an item, you have provide **ForumName** (table Primary key), **Subject** (table Sort key) and **CreationDate** (Local Secondary Index Sort Key). Also add **Body** attribute:
2. Add first item and save:



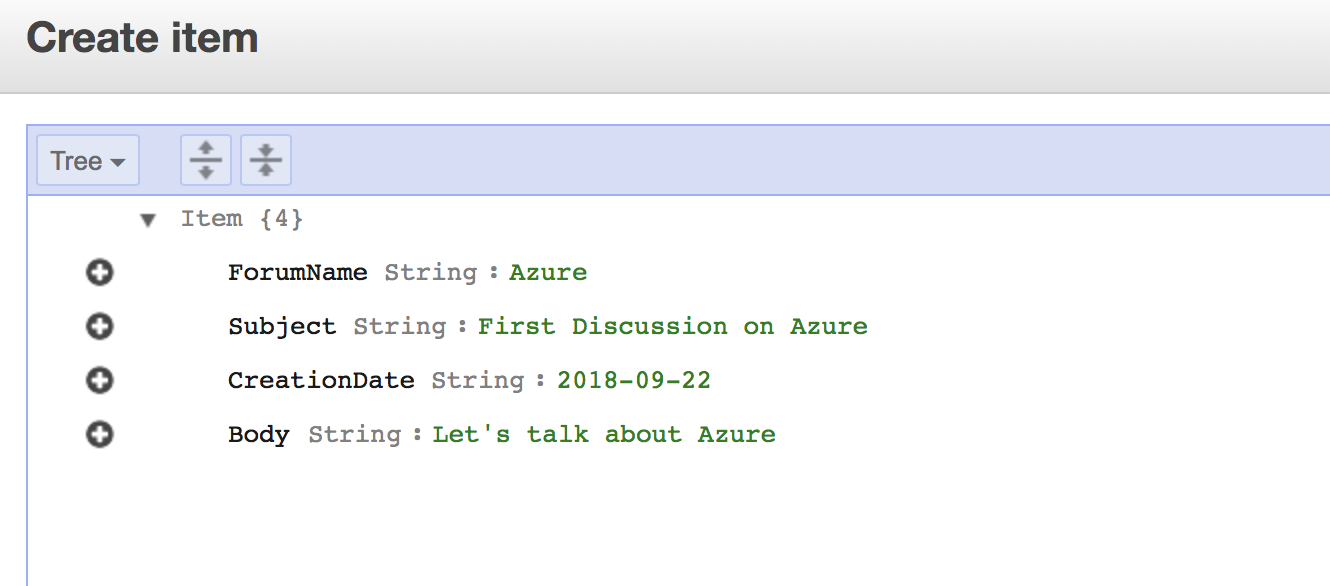
1. Add second item and save (use Duplicate feature for faster creation):



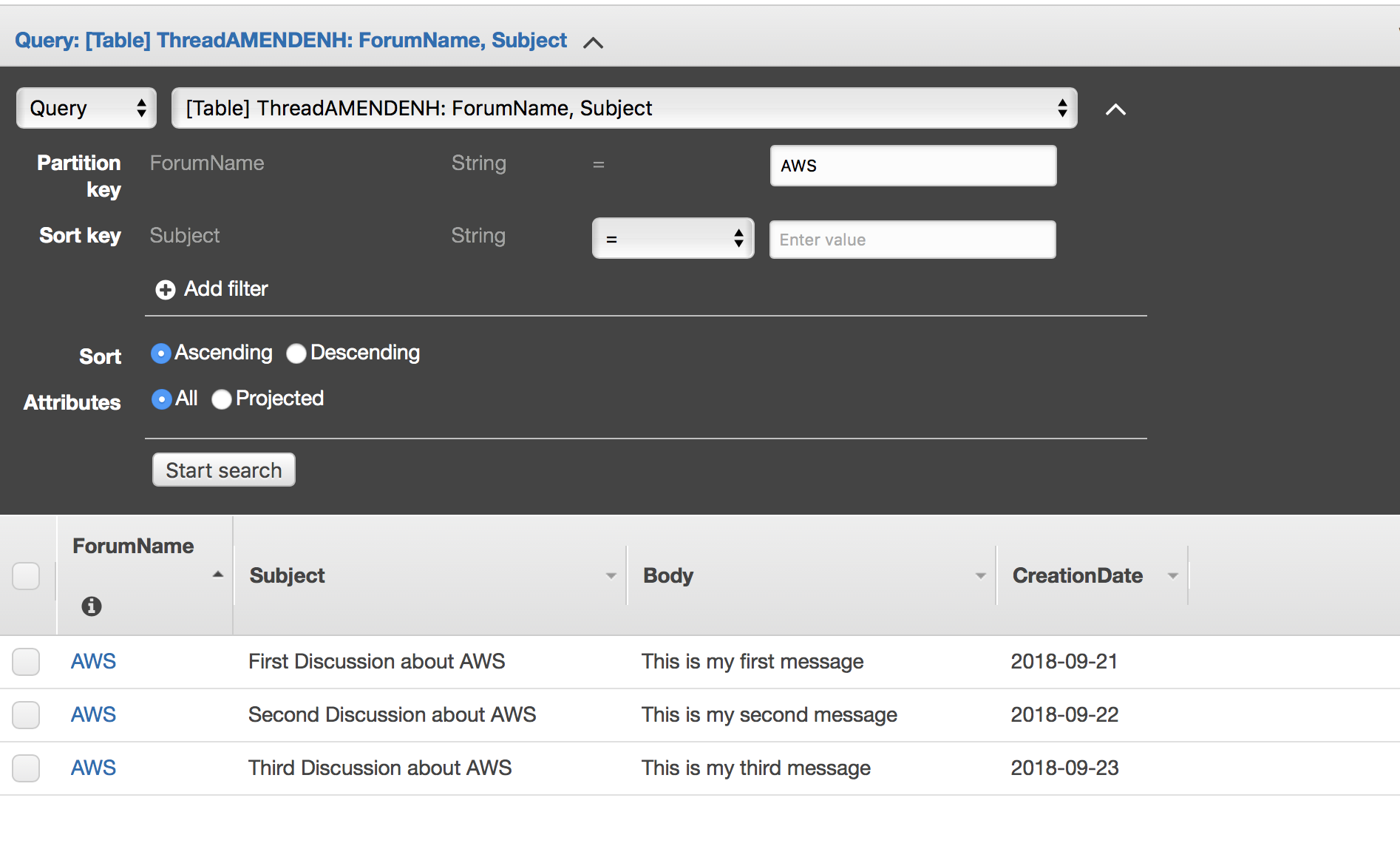
1. Add third item and save:



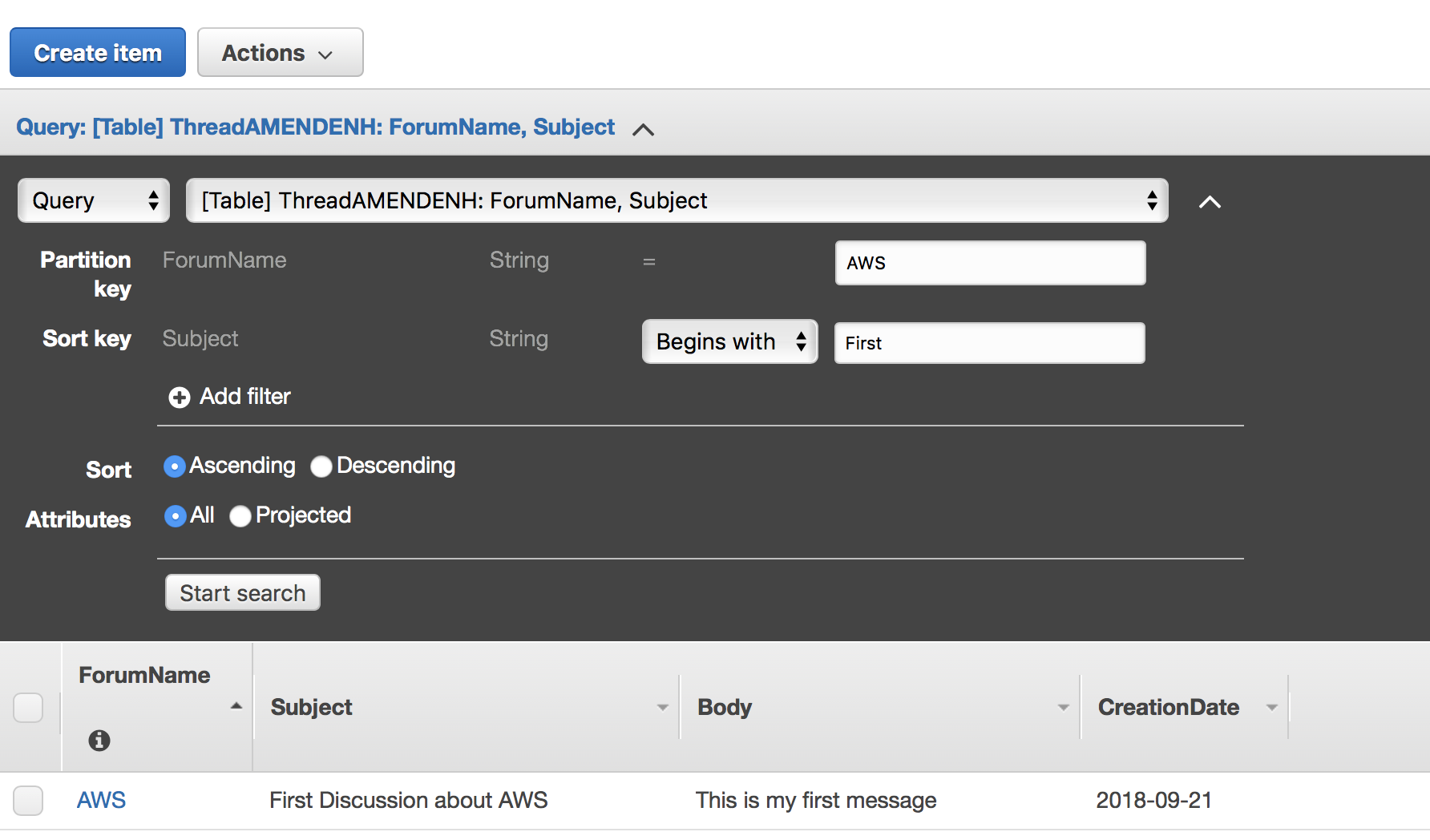
1. Add forth item and save:

****

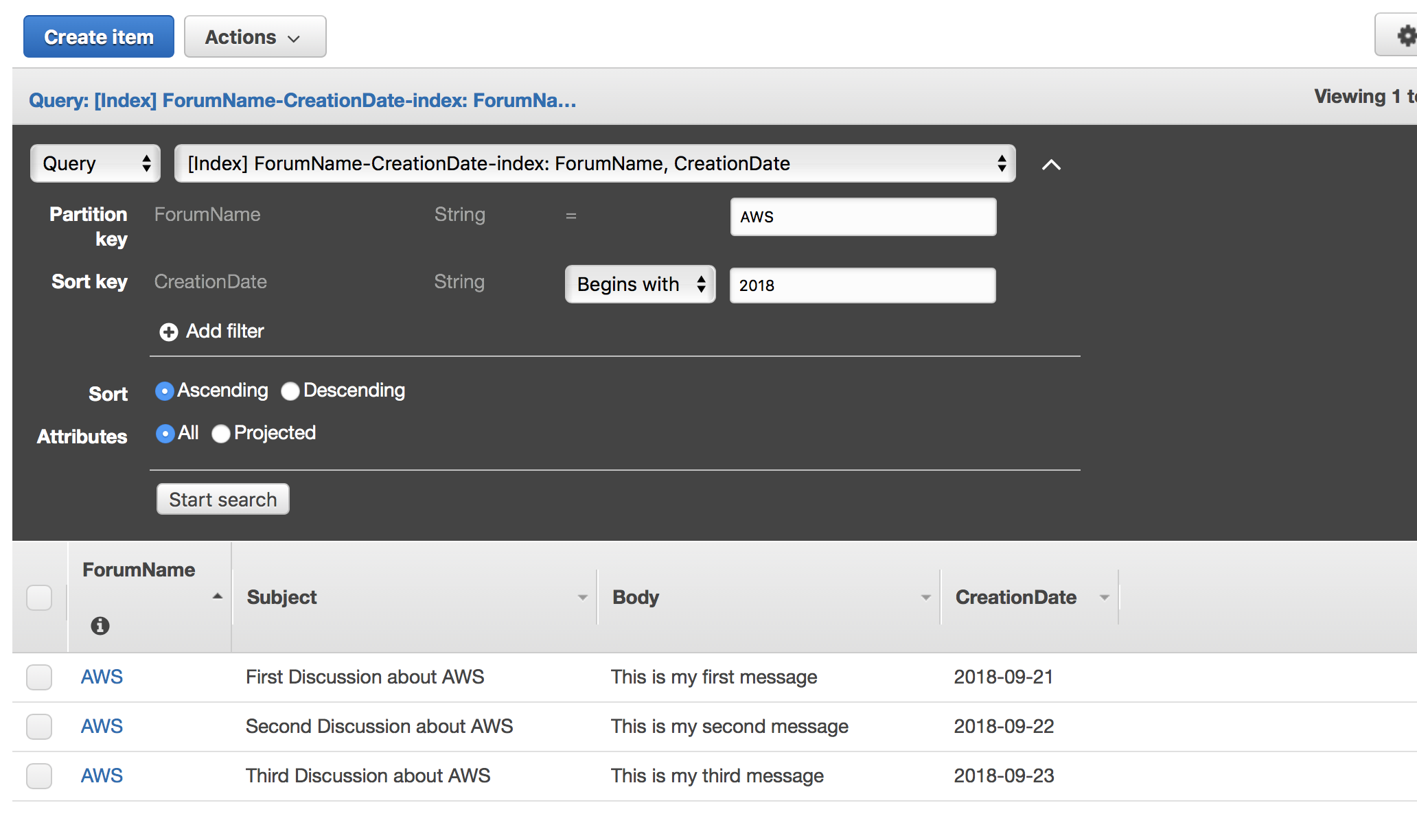
1. **Scan and Query items**. Remember that **Scan** operation looks at every item in the table and returns all the data attributes for all. **Query** finds items using primary key and optionally sort key to refine your search.
2. **ThreadYOURNETID** table has Local index on it (**CreationDate**) along with Partition and Sort key. Let’s Query **ThreadYOURNETID** table and enter **AWS** for Partition Key (Azure forum name should be omitted). Notice how query results are sorted by Sort Key:

****

1. Let’s specify Sort keys with **Subject Begin With = First:**



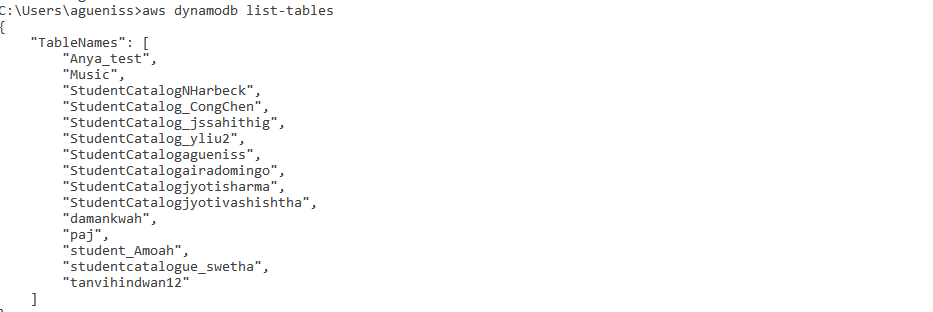
1. Let’s use **Local Secondary Index** by switching to Index Name. Chose Forum Name = **AWS** and **CreationDate** Begins with = 2018 as follows:



**Exercise 3 (optional)**

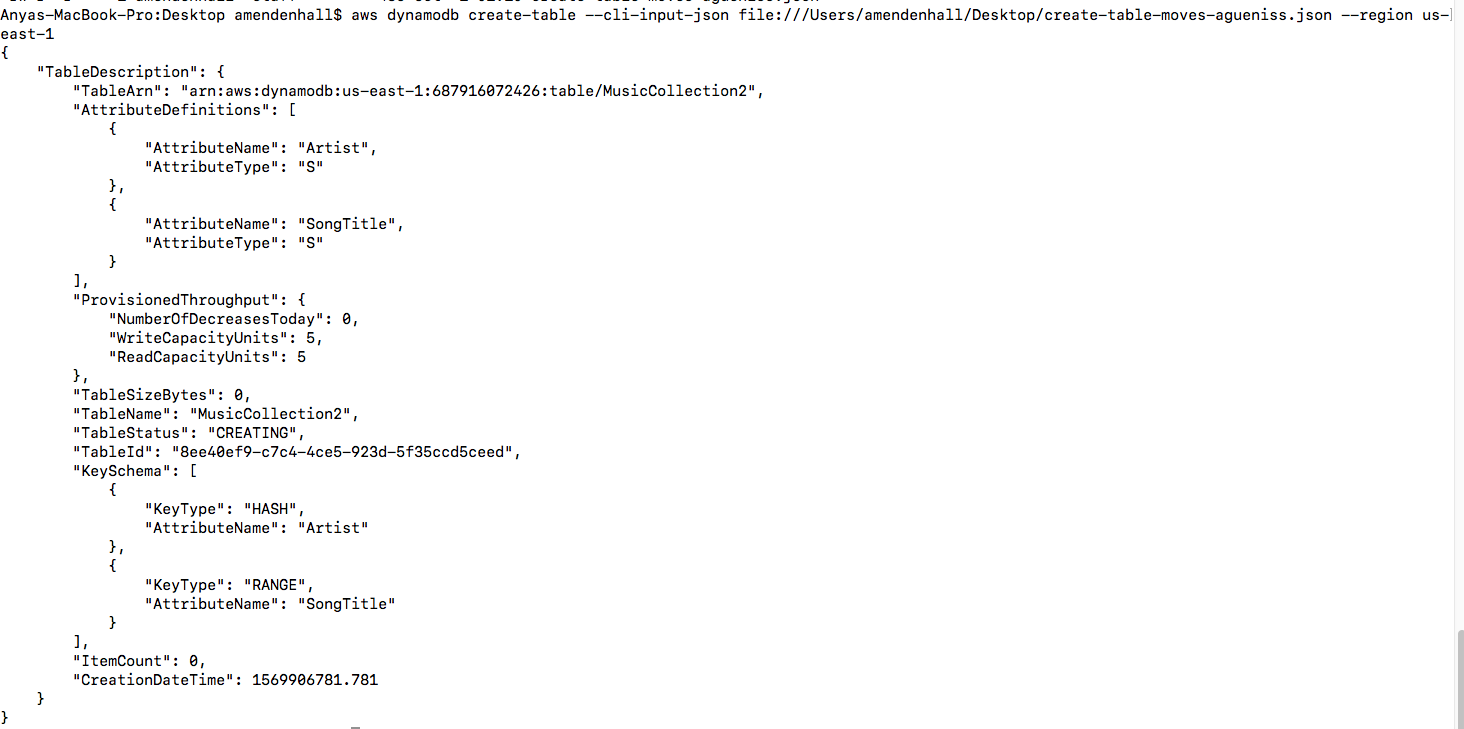
**Learning Objective**

1. Install and configure AWS CLI
2. List DynamoDB tables
3. Import table into DynamoDB via AWS CLI
4. Observe data in your DynamoDB database through the AWS console
5. Follow AWS CLI configuration instructions posted on Blackboard (separate for Windows and Mac).
6. List DynamoDB tables:



1. Download create-table-movies.json and **rename it to create-table-movies-YOURNETID.json** from Blackboard to your Desktlop and execute create-table command as follows (**SPECIFY FULL PATH WHERE YOUR JSON FILE IS LOCATED AND ADD a unique number to your table name inside json file as we cannot have tables in the same AWS account with the same name**:

aws dynamodb create-table --cli-input-json file://create-table-movies-**yournetid**.json --region us-east-1



1. Login to AWS console and observe your table